

Functionally-fitted explicit pseudo two-step Runge-Kutta-Nyström methods

N. S. Hoang*

Abstract

A general class of functionally-fitted explicit pseudo two-step Runge-Kutta-Nyström (FEPTRKN) methods for solving second-order initial value problems has been studied. These methods can be considered generalized explicit pseudo two-step Runge-Kutta-Nyström (EPTRKN) methods. We proved that an s -stage FEPTRKN method has step order $p = s$ and stage order $r = s$ for any set of distinct collocation parameters $(c_i)_{i=1}^s$. Supperconvergence for the accuracy orders of these methods can be obtained if the collocation parameters $(c_i)_{i=1}^s$ satisfy some orthogonality conditions. We proved that an s -stage FEPTRKN method can attain accuracy order $p = s + 3$. Numerical experiments have shown that the new FEPTRKN methods work better than do EPTRKN methods on problems whose solutions can be well approximated by the functions in bases on which these FEPTRKN methods are developed.

Keywords. functionally-fitted, generalized collocation, variable coefficients, two-step explicit RKN, nonstiff ODEs.

1 Introduction

Consider the initial value problem

$$y''(t) = f(t, y(t)), \quad y(0) = y_0, \quad y'(0) = y'_0, \quad t \in [t_0, t_0 + T] \quad (1)$$

where $f : [t_0, t_0 + T] \times \mathbb{R} \rightarrow \mathbb{R}$ and $f(t, y)$ is continuous with respect to t and satisfies a Lipschitz condition with respect to y . For simplicity of presentation we state equation (1) in scalar form. However, with a change of notation, the discussion also holds when equation (1) is in vector form. One of the approaches to solve numerically equation (1) is to rewrite the equation as a system of first-order ODEs and use numerical methods to solve this system. There are many methods for solving systems of first-order ODEs (see, e.g., [1], [10]). The disadvantage of this approach is that the sizes of the obtained systems are twice as large as the sizes of the original systems. Also, this approach does not take advantage of the special form of equation (1). It has been of interest to develop methods to solve equation (1) without rewriting it as a system of first-order ODEs.

Numerical methods for solving equation (1) directly have been developed extensively in the literature (see, e.g., [2], [3], [5], [6], and [18]). Among direct methods for solving (1), Runge-Kutta-Nyström (RKN) methods are preferable. An s -stage RKN method is defined by its Butcher-tableau as follows

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline \mathbf{b}^T & \\ \mathbf{d}^T & \end{array}, \quad \mathbf{A} = [a_{ij}] \in \mathbb{R}^{s \times s}, \quad \mathbf{b} = (b_1, \dots, b_s)^T, \quad \mathbf{d} = (d_1, \dots, d_s)^T, \quad \mathbf{c} = (c_1, \dots, c_s)^T.$$

*Department of Mathematics, University of West Georgia, nhoang@westga.edu

Given y_n and y'_n , the approximations of $y(t_n)$ and $y'(t_n)$ at the n -th step, the approximations of $y(t_{n+1})$ and $y'(t_{n+1})$ at the $(n+1)$ -th step defined by the s -stage RKN method $(\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{d})$ are

$$y_{n+1} = y_n + hy'_n + h^2 \sum_{j=1}^s b_j f(t_n + c_j h, Y_{n,j}), \quad (2)$$

$$y'_{n+1} = y'_n + h \sum_{j=1}^s d_j f(t_n + c_j h, Y_{n,j}), \quad (3)$$

$$Y_{n,i} = y_n + c_i h y'_n + h^2 \sum_{j=1}^s a_{ij} f(t_n + c_j h, Y_{n,j}), \quad i = 1, \dots, s. \quad (4)$$

For an implicit RKN method, the matrix \mathbf{A} is nonsingular and (4) is a system of nonlinear equations. In this case it requires special techniques such as Newton methods or fixed-point iterations to solve equation (4). For an explicit RKN method, \mathbf{A} is strictly lower-triangular and $c_1 = 0$. The stage values $(Y_{n,i})_{i=1}^s$ can be easily computed from

$$Y_{n,1} = y_n, \quad Y_{n,i} = y_n + h c_i y'_n + h^2 \sum_{j=1}^{i-1} a_{ij} f(t_n + c_j h, Y_{n,j}), \quad i = 2, \dots, s. \quad (5)$$

Recently, much work has been devoted to the study of functionally-fitted methods. These methods are developed to integrate an ODE exactly if the solution of the ODE is a linear combination of some certain basis functions (see, e.g., [2], [7], [8], [12], [13], [14], [17]). For example, trigonometric methods have been developed to solve periodic or nearly periodic problems (see, e.g., [7], [11], [17]). Numerical experiments have shown that trigonometrically-fitted methods are superior to classical Runge-Kutta methods for solving ODEs whose solutions are periodic or nearly periodic functions with known frequencies (see, e.g., [8], [11], [12], [13], [14], [16], [17]).

In [14], a general class of functionally-fitted explicit pseudo two-step Runge-Kutta (FEPTRK) methods was developed. These methods can be considered generalized explicit pseudo two-step Runge-Kutta (EPTRK) methods. Although EPTRK methods were originally developed for parallel computers (cf. [4]), it was shown in [14] that (F)EPTRK methods work better than one of the most frequently used Runge-Kutta methods the DOPRI45 method even in sequential computing environments on non-stiff problems. In [3], a class of explicit pseudo two-step Runge-Kutta-Nyström (EPTRKN) methods for solving non-stiff problem (1) was developed. In particular, it was proved in [3] that an s -stage EPTRKN method has step order $p = s$ for any set of collocation points $(c_i)_{i=1}^s$. For super-convergence it was shown in [3] that an s -stage EPTRKN can attain step accuracy order up to $p = s + 2$ if the collocation parameters $(c_i)_{i=1}^s$ satisfy some orthogonality conditions. The accuracy study in [3] was done by using Taylor series expansion technique. It was shown in [5] that these methods were more efficient than some of the most commonly used methods for solving non-stiff large-scale systems of ODEs. When solving non-stiff problems it is the accuracy not the stability that controls the stepsizes of numerical methods.

In this paper, we study a general class of functionally-fitted explicit pseudo Runge-Kutta-Nyström (FEPTRKN) methods. Similar to FEPTRK methods in [14], these methods have the advantage of integrating an ODE exactly if the solution of the ODE is a linear combination of functions in bases on which the FEPTRKN methods are developed. Using a similar collocation framework as in [14], we have shown in this paper that an s -stage FEPTRKN method has stage order $p = s$ for any set of collocation parameters $(c_i)_{i=1}^s$. When the basis functions are polynomials, we recover the results for EPTRKN methods in [3]. Moreover, we proved in this paper that an s -stage FEPTRKN can attain step order $p = s + 3$. A consequence of this result is that an s -stage EPTRKN method can have step order $p = s + 3$. This superconvergence result is new and allows us to construct methods of higher accuracy orders than those in [3, 6] for a given number of stage s . Using this superconvergence result,

we have constructed 4-stage, 5-stage, and 6-stage FEPTTRKN methods having accuracy order 7, 8, and 9, respectively. Numerical experiments with the new methods have shown that these methods attain accuracy orders as indicated by the theory. Moreover, these FEPTTRKN methods have performed much better than the corresponding EPTRKN methods when the basis functions are suitably chosen.

2 Explicit pseudo two-step RKN (EPTRKN) methods

The iteration scheme of a conventional RKN method based on (2)–(4) can be represented compactly as

$$\mathbf{Y}_n = \mathbf{e}y_n + h\mathbf{c}y'_n + h^2\mathbf{A}f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n) \in \mathbb{R}^s, \quad (6a)$$

$$y_{n+1} = y_n + hy'_n + h^2\mathbf{b}^T f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n) \in \mathbb{R}, \quad (6b)$$

$$y'_{n+1} = y'_n + h\mathbf{d}^T f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n) \in \mathbb{R}, \quad (6c)$$

where $\mathbf{Y}_n := (Y_{n,1}, \dots, Y_{n,s})^T$ and $f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n) := (f(t_n + c_1h, Y_1), \dots, f(t_n + c_sh, Y_s))^T$. For implicit RKN methods, one has to solve nonlinear equation (6a) for the stage vector \mathbf{Y}_n and this requires extra computation time. It should be noted that all classical collocation RKN methods are implicit (cf. [18]). It is well known that implicit RK(N) methods should only be used for solving stiff problems. For non-stiff problems, explicit methods are computationally cheaper as the stage vector \mathbf{Y}_n can be computed easily.

In [3] Cong defined the iteration scheme of an s -stage explicit pseudo two-step RKN (EPTRKN) method as

$$y_{n+1} = y_n + hy'_n + h^2\mathbf{b}^T f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n) \in \mathbb{R}, \quad (7a)$$

$$y'_{n+1} = y'_n + h\mathbf{d}^T f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n) \in \mathbb{R}, \quad (7b)$$

$$\mathbf{Y}_{n+1} = \mathbf{e}y_{n+1} + h\mathbf{c}y'_{n+1} + h^2\mathbf{A}f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n) \in \mathbb{R}^s, \quad (7c)$$

where, again, $y_n \approx y(t_n)$, $y'_n \approx y'(t_n)$, and $\mathbf{Y}_n = (Y_{n,1}, \dots, Y_{n,s})^T \approx y(\mathbf{e}t_n + \mathbf{c}h) = (y(t_n + c_1h), \dots, y(t_n + c_sh))^T$. The advantage of EPTRKN methods over implicit RKN methods is: the stage vector \mathbf{Y}_{n+1} in equation (7c) can be computed explicitly using the values of y_n , y'_n , and \mathbf{Y}_n from the previous step. The scheme needs s sufficiently accurate starting values to define \mathbf{Y}_0 , but these can be obtained by a conventional method. Since the components of $f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n)$ can be evaluated independently in parallel computing environments, EPTRKN methods are ideally suited for parallel computers. Note that on parallel computing environments, EPTRKN methods use only one function evaluation of $f(t_n + c_ih, Y_{n,i})$ per step.

3 Functionally-fitted explicit pseudo two-step RKN (FEPTTRKN) methods

Let us first give a definition of functionally-fitted explicit pseudo two-step RKN (FEPTTRKN) methods. This requires choosing a set of basis functions $\{u_i(t)\}_{i=1}^s$ that are sufficiently smooth and linearly independent and satisfy the integration scheme (7) exactly.

Definition 3.1 (Functionally-fitted EPTRKN) *An s -stage EPTRKN method is a functionally-fitted (or generalized collocation) explicit pseudo two-step RKN (FEPTTRKN) method with respect to the basis functions $\{u_i(t)\}_{i=1}^s$ if the following relations are satisfied for all $i = 1, \dots, s$:*

$$\begin{aligned} u_i(t+h) &= u_i(t) + hu'_i(t) + h^2\mathbf{b}^T(t, h)u''_i(\mathbf{e}t + \mathbf{c}h), \\ u'_i(t+h) &= u'_i(t) + h\mathbf{d}^T(t, h)u''_i(\mathbf{e}t + \mathbf{c}h), \\ u_i(\mathbf{e}t + \mathbf{c}h + \mathbf{e}h) &= \mathbf{e}u_i(t+h) + h\mathbf{c}u'_i(t+h) + h^2\mathbf{A}(t, h)u''_i(\mathbf{e}t + \mathbf{c}h). \end{aligned} \quad (8)$$

Here, $\mathbf{b}^T(t, h)$ and $\mathbf{d}^T(t, h)$ denote the transposes of $\mathbf{b}(t, h)$ and $\mathbf{d}(t, h)$, respectively. This immediately yields linear systems to solve for \mathbf{A} , \mathbf{b} , and \mathbf{d} . These coefficients generally depend on both t and h . The parameters $(c_i)_{i=1}^s$ are distinct and will be chosen later. By construction, FEPTTRKN methods are explicit.

3.1 Collocation condition

It is clear that not all bases of functions $\{u_i(t)\}_{i=1}^s$ satisfy (8), i.e., there are some sets of functions $\{u_i(t)\}_{i=1}^s$ for which one cannot find \mathbf{A} , \mathbf{b} , and \mathbf{d} so that condition (8) holds for these $\{u_i(t)\}_{i=1}^s$. The *collocation condition* defined below guarantees the existence of \mathbf{A} , \mathbf{b} and \mathbf{d} satisfying condition (8), and, therefore, guarantees the existence of the corresponding FEPTTRKN method.

Definition 3.2 (Collocation condition) *A set of sufficiently smooth functions $\{u_1(t), u_2(t), \dots, u_s(t)\}$ is said to satisfy the collocation condition if for any given $t \in [0, T]$, the matrix*

$$\mathbf{F}(t, h) := \begin{pmatrix} u_1''(et + ch), u_2''(et + ch), \dots, u_s''(et + ch) \end{pmatrix}$$

is nonsingular for almost every h in the interval $[0, h_{\max}]$, $h_{\max} = \text{const} > 0$.

Remark 3.1 Equations in (8) can be written as

$$\begin{aligned} \begin{pmatrix} u_1(t+h) - u_1(t) - hu_1'(t), \dots, u_s(t+h) - u_s(t) - hu_s'(t) \end{pmatrix} &= h^2 \mathbf{b}^T(t, h) \mathbf{F}(t, h), \\ \begin{pmatrix} u_1'(t+h) - u_1'(t), \dots, u_s'(t+h) - u_s'(t) \end{pmatrix} &= h \mathbf{d}^T(t, h) \mathbf{F}(t, h), \\ \begin{pmatrix} v_1(t, h), \dots, v_s(t, h) \end{pmatrix} &= h^2 \mathbf{A}(t, h) \mathbf{F}(t, h), \end{aligned} \quad (9)$$

where

$$v_i(t, h) := u_i(et + eh + ch) - eu_i(t+h) - hcu_i'(t+h), \quad i = 1, \dots, s.$$

Hence, the existence of the coefficients $\mathbf{A}(t, h)$, $\mathbf{d}(t, h)$, $\mathbf{b}(t, h)$ at $t \in [t_0, t_0 + T]$ and $h > 0$ is guaranteed if $\mathbf{F}(t, h)$ is nonsingular.

Remark 3.2 If $\{u_i(t)\}_{i=1}^s$ satisfies the collocation condition, then the linear function $\alpha + \beta t$ is not in the linear space $\text{Span}\{u_1, u_2, \dots, u_s\}$ for any constants α and β . In other words, there do not exist constants $(\alpha_i)_{i=1}^s$ so that $\alpha_1 u_1(t) + \alpha_2 u_2(t) + \dots + \alpha_s u_s(t) = \alpha + \beta t$ for some constants α and β . In particular, none of the functions $\{u_i(t)\}_{i=1}^s$ is a linear function.

Remark 3.3 When $\{u_1(t), u_2(t), \dots, u_s(t)\} = \{t^2, t^3, \dots, t^{s+1}\}$, we have

$$\mathbf{F}(t, h) = \begin{pmatrix} 2e, 6(et + ch), \dots, (s+1)s(et + ch)^{s-1} \end{pmatrix}.$$

Subtracting between columns in the matrix $\mathbf{F}(t, h)$, we see that

$$\det \mathbf{F}(t, h) = \begin{vmatrix} 2 & 6c_1 h & \cdots & (s+1)s(c_1 h)^{s-1} \\ 2 & 6c_2 h & \cdots & (s+1)s(c_2 h)^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 2 & 6c_s h & \cdots & (s+1)s(c_s h)^{s-1} \end{vmatrix} = s!(s+1)!h^{\frac{s(s-1)}{2}} \begin{vmatrix} 1 & c_1 & \cdots & c_1^{s-1} \\ 1 & c_2 & \cdots & c_2^{s-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_s & \cdots & c_s^{s-1} \end{vmatrix}.$$

Observe that $\det \mathbf{F}(t, h)$ is a multiple of the determinant of a Vandermonde matrix and, therefore, is nonzero if $h \neq 0$ and $(c_i)_{i=1}^s$ are distinct. Hence, the collocation condition is always satisfied when the fitting functions $\{u_i(t)\}_{i=1}^s$ are the monomials.

The practical implication of the proposed collocation condition is that for any given t we can control the stepsize h to get a nonsingular $\mathbf{F}(t, h)$ and then solve for \mathbf{A}, \mathbf{b} , and \mathbf{d} . For a given set of functions $\{u_1(t), \dots, u_s(t)\}$ which satisfies the collocation condition and a given $t \in [t_0, t_0 + T)$, there will be at most countably many values of h such that $\mathbf{F}(t, h)$ is singular. Thus, we have the following result.

Theorem 3.1 *The coefficients of a FEPTTRKN method with respect to a set of basis functions that satisfy the collocation condition are uniquely determined almost everywhere on the integration domain.*

3.2 The collocation solution

Let $(c_i)_{i=1}^s$ be distinct and $\{u_i\}_{i=1}^s$ be a basis satisfying the collocation condition. Define

$$\mathbf{H} := \text{Span}\{1, t, u_1, \dots, u_s\} := \left\{ \alpha_0 + a_0 t + \sum_{i=1}^s a_i u_i(t) \mid \alpha_0 \in \mathbb{R}, a_i \in \mathbb{R}, i = 0, \dots, s \right\}.$$

Given y_n, y'_n , and $f(t_n + c_i h, Y_{n,i})$, $i = 1, \dots, s$, we call $u(t)$ the *collocation solution* if $u(t) \in \mathbf{H}$ and the following equations hold

$$\begin{aligned} u(t_n) &= y_n, \\ u'(t_n) &= y'_n, \\ u''(t_n + c_i h) &= f(t_n + c_i h, Y_{n,i}), \quad i = 1, \dots, s. \end{aligned} \tag{10}$$

If such a $u(t)$ exists, the numerical solution, derivative of the solution, and the stage values at the $(n+1)$ -th step are defined by

$$y_{n+1} := u(t_{n+1}), \quad y'_{n+1} := u'(t_{n+1}), \quad Y_{n+1,i} := u(t_{n+1} + c_i h), \quad i = 1, \dots, s, \quad t_{n+1} = t_n + h. \tag{11}$$

Equations (10) and (11) can be called a generalized collocation method for integrating equation (1). When y_n, y'_n , and $\mathbf{Y}_n = (Y_{n,1}, \dots, Y_{n,s})^T$ are known, $u(t)$ can be constructed explicitly through a direct interpolation involving y_n, y'_n , and $f(\mathbf{e}t_n + \mathbf{c}h, \mathbf{Y}_n)$. The existence of such collocation solution is made more precise as follows.

Lemma 3.2 *Suppose that the $s + 2$ values $z_0, z'_0, z_1, z_2, \dots, z_s$ are given and the pair (t_n, h) is such that $\mathbf{F}(t_n, h)$ is nonsingular, then there exists an interpolation function $\varphi \in \mathbf{H}$ such that*

$$\varphi(t_n) = z_0, \quad \varphi'(t_n) = z'_0, \quad \varphi''(t_n + c_i h) = z_i, \quad i = 1, \dots, s. \tag{12}$$

Proof. Since $\varphi \in \mathbf{H}$, it can be represented in the form

$$\varphi(t) = \alpha_0 + a_0 t + \sum_{i=1}^s a_i u_i(t).$$

Let $t_{n,i} := t_n + c_i h$, $i = 1, \dots, s$. Equation (12) can be written as

$$\begin{pmatrix} 1 & t_n & u_1(t_n) & u_2(t_n) & \cdots & u_s(t_n) \\ 0 & 1 & u'_1(t_n) & u'_2(t_n) & \cdots & u'_s(t_n) \\ 0 & 0 & u''_1(t_{n,1}) & u''_2(t_{n,1}) & \cdots & u''_s(t_{n,1}) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & u''_1(t_{n,s}) & u''_2(t_{n,s}) & \cdots & u''_s(t_{n,s}) \end{pmatrix} \begin{pmatrix} \alpha_0 \\ a_0 \\ a_1 \\ \vdots \\ a_s \end{pmatrix} = \begin{pmatrix} z_0 \\ z'_0 \\ z_1 \\ \vdots \\ z_s \end{pmatrix}. \tag{13}$$

Since the matrix in the left-hand side of (13) has the same determinant as $\mathbf{F}(t_n, h)$, the constants $(a_i)_{i=0}^s$ and α_0 are uniquely determined from linear system (13) under the assumption that $\mathbf{F}(t_n, h)$ is nonsingular. Thus, the function $\varphi(t) \in \mathbf{H}$ satisfying (12) is determined uniquely. \square

Theorem 3.3 *The generalized collocation method (10)–(11) is equivalent to the s -stage FEPTRKN method with coefficients $(\mathbf{c}, \mathbf{A}(t_n, h), \mathbf{b}(t_n, h), \mathbf{d}(t_n, h))$.*

Proof. It follows from Lemma 3.2 that there exists a unique interpolation function $u(t) \in \mathbf{H}$ such that

$$u(t_n) = y_n, \quad u'(t_n) = y'_n, \quad u''(t_n + c_i h) = f(t_n + c_i h, Y_{n,i}), \quad i = 1, \dots, s. \quad (14)$$

Thus, $u(t)$ satisfies (10) and is the collocation solution being searched for. To complete the proof, we have to prove that if we use $u(t)$ to generate the quantities in (11), then the following relations that define the FEPTRKN method hold true:

$$\begin{aligned} y_{n+1} &= y_n + h y'_n + h^2 \mathbf{b}^T(t_n, h) f(et_n + \mathbf{c}h, \mathbf{Y}_n), \\ y'_{n+1} &= y'_n + h \mathbf{d}^T(t_n, h) f(et_n + \mathbf{c}h, \mathbf{Y}_n), \\ \mathbf{Y}_{n+1} &= \mathbf{e} y_{n+1} + h \mathbf{c} y'_{n+1} + h^2 \mathbf{A}(t_n, h) f(et_n + \mathbf{c}h, \mathbf{Y}_n). \end{aligned} \quad (15)$$

Since $u(t) \in \mathbf{H}$, it has the following representation

$$u(t) = \alpha_0 + a_0 t + \sum_{i=1}^s a_i u_i(t). \quad (16)$$

From the definition of a FEPTRKN method in (8), the coefficients $(\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{d})$ satisfy

$$u_i(t_{n+1}) = u_i(t_n) + h u'_i(t_n) + h^2 \mathbf{b}^T(t_n, h) u''_i(et_n + \mathbf{c}h), \quad (17a)$$

$$u'_i(t_{n+1}) = u'_i(t_n) + h \mathbf{d}^T(t_n, h) u''_i(et_n + \mathbf{c}h), \quad (17b)$$

$$u_i(et_n + \mathbf{e}h + \mathbf{c}h) = \mathbf{e} u_i(t_n + h) + \mathbf{c} h u'_i(t_n + h) + h^2 \mathbf{A}(t_n, h) u''_i(et_n + \mathbf{c}h), \quad (17c)$$

for all $i = 1, \dots, s$. From equation (17a) and the fact that $u(t)$ is a linear combination of functions $u_i(t)$, 1, and t (cf. (16)), we obtain

$$u(t_{n+1}) = u(t_n) + h u'(t_n) + h^2 \mathbf{b}^T(t_n, h) u''(et_n + \mathbf{c}h). \quad (18)$$

Equation (11), equation (14), and equation (18) imply that

$$y_{n+1} = u(t_{n+1}) = y_n + h y'_n + h^2 \mathbf{b}^T(t_n, h) f(et_n + \mathbf{c}h, \mathbf{Y}_n).$$

Thus, the first equation in (15) holds. The other equations in (15) are obtained similarly. Theorem 3.3 is proved. \square

4 Accuracy and stability properties

4.1 Accuracy order

The stage order and step order of a FEPTRKN method are defined as follows (cf. [3]).

Definition 4.1 *A FEPTRKN method is said to have step order $p = \min\{p_1, p_2\}$ and stage order $r = \min\{p_1, p_2, p_3\}$ if*

$$\begin{aligned} y(t_{n+1}) - y_{n+1} &= O(h^{p_1+1}), \\ y'(t_{n+1}) - y'_{n+1} &= O(h^{p_2+1}), \\ y(et_{n+1} + \mathbf{c}h) - \mathbf{Y}_{n+1} &= O(h^{p_3+1}), \end{aligned}$$

given that $y_n = y(t_n)$, $y'_n = y'(t_n)$, and $y(et_n + \mathbf{c}h) - \mathbf{Y}_n = O(h^{p_3+1})$.

Remark 4.1 We recall this handy result from [12, Remark 3.1]. If a function $f(t) \in C^{m+n}([t_0, t_0 + T])$ satisfies $f(t_i) = 0, i = 1, \dots, n$, then there exists $g(t) \in C^m([t_0, t_0 + T])$ such that $f(t) = g(t) \prod_{i=1}^n (t - t_i)$.

Theorem 4.1 An s -stage FEPTRKN method has stage order $r = s$ and step order at least $p = s$.

Proof. Without loss of generality we assume that $t_n = 0$. Let $u(t) \in \mathbf{H}$ be the collocation solution satisfying

$$u(t_n) = y_n, \quad u'(t_n) = y'_n, \quad u''(\mathbf{c}h) = f(\mathbf{c}h, \mathbf{Y}_n). \quad (19)$$

This collocation solution exists by Lemma 3.2 and we have

$$y_{n+1} = u(t_{n+1}), \quad y'_{n+1} = u'(t_{n+1}), \quad \mathbf{Y}_{n+1} = u(\mathbf{e}h + \mathbf{c}h).$$

From the equation $y''(t) = f(t, y(t)), \forall t \in [t_0, t_0 + T]$, the third equation in (19), the local assumption $\mathbf{Y}_n - y(\mathbf{c}h) = O(h^{s+1})$, and Taylor expansions of $f(t, y)$ with respect to y , one gets

$$u''(\mathbf{c}h) - y''(\mathbf{c}h) = f(\mathbf{c}h, \mathbf{Y}_n) - f(\mathbf{c}h, y(\mathbf{c}h)) = O(h^{s+1}). \quad (20)$$

Let

$$R(t) := u(t) - y(t), \quad 0 \leq t \leq t_0 + T.$$

From equations (19) and (20) one gets

$$R(0) = 0, \quad R'(0) = 0, \quad R''(c_i h) = u''(c_i h) - y''(c_i h) = O(h^{s+1}), \quad i = 1, \dots, s.$$

Since $R''(t)$ is sufficiently smooth and $R''(c_i h) = O(h^{s+1}), i = 1, \dots, s$, there exists a sufficiently smooth function $w(t)$ such that

$$R''(t) = w(t) + O(h^{s+1}), \quad w(c_i h) = 0, \quad i = 1, \dots, s.$$

Since $w(c_i h) = 0, i = 1, \dots, s$, it follows from Remark 4.1 that there exists a sufficiently smooth function $g(t)$ such that

$$w(t) = g(t) \prod_{i=1}^s (t - c_i h).$$

Thus,

$$R''(t) = g(t) \prod_{i=1}^s (t - c_i h) + O(h^{s+1}). \quad (21)$$

This and the relation $R'(0) = 0$ imply

$$R'(x) = \int_0^x R''(t) dt = \int_0^x \left(g(t) \prod_{i=1}^s (t - c_i h) + O(h^{s+1}) \right) dt.$$

Letting $x = h$ in the above equality and using the substitution $t = \xi h$, we get

$$R'(h) = h^{s+1} \int_0^1 g(\xi h) \prod_{i=1}^s (\xi - c_i) d\xi + \int_0^h O(h^{s+1}) dt. \quad (22)$$

This and the relation $\int_0^h O(h^{s+1}) dt = O(h^{s+2})$ imply

$$y'_{n+1} - y'(t_{n+1}) = u'(t_{n+1}) - y'(t_{n+1}) = R'(h) = O(h^{s+1}). \quad (23)$$

From equation (21) and the equalities $R(0) = 0$ and $R'(0) = 0$, one gets

$$R(x) = \int_0^x \int_0^t R''(\xi) d\xi dt = \int_0^x \int_0^t \left(g(\xi) \prod_{i=1}^s (\xi - c_i h) + O(h^{s+1}) \right) d\xi dt.$$

Substituting $x := \omega h$ in the above equality and using substitutions, we obtain

$$u(\omega h) - y(\omega h) = R(\omega h) = h^{s+2} \int_0^\omega \int_0^t g(\xi h) \prod_{i=1}^s (\xi - c_i) d\xi dt + \int_0^{\omega h} \int_0^t O(h^{s+1}) d\xi dt. \quad (24)$$

This and the relation $\int_0^{\omega h} \int_0^t O(h^{s+1}) d\xi dt = O(h^{s+3})$, $\omega = \text{const}$, imply

$$\mathbf{Y}_{n+1} - y(\mathbf{e}h + \mathbf{c}h) = u(\mathbf{e}h + \mathbf{c}h) - y(\mathbf{e}h + \mathbf{c}h) = O(h^{s+2}), \quad (25)$$

$$y_{n+1} - y(h) = u(h) - y(h) = O(h^{s+2}). \quad (26)$$

Equations (23), (25), (26), and Definition 4.1 imply that the method has step order $p = s$ and stage order $r = s$.

Theorem 4.1 is proved. \square

Remark 4.1 From (25) we have

$$\mathbf{Y}_n - y(\mathbf{e}t_n + \mathbf{c}h) = O(h^{s+2}), \quad n = 1, 2, \dots. \quad (27)$$

Thus, if \mathbf{Y}_0 is computed with a local accuracy order of $s + 2$ at the initial step, then we have

$$\mathbf{Y}_n - y(\mathbf{e}t_n + \mathbf{c}h) = O(h^{s+2}), \quad n = 0, 1, \dots. \quad (28)$$

4.2 Superconvergence

Theorem 4.2 *If an s -stage FEPTRKN method has collocation parameters $(c_i)_{i=1}^s$ that satisfy*

$$\int_0^1 \prod_{i=1}^s (\xi - c_i) d\xi = 0, \quad (29)$$

then the method has step order $p = s + 1$ and stage order $r = s + 1$.

Proof. From equation (22) and the relation $\int_0^h O(h^{s+1}) dt = O(h^{s+2})$, we have

$$y'_{n+1} - y'(t_{n+1}) = u'(t_{n+1}) - y'(t_{n+1}) = R'(h) = h^{s+1} \int_0^1 g(\xi h) \prod_{i=1}^s (\xi - c_i) d\xi + O(h^{s+2}).$$

Use the Taylor expansion $g(\xi h) = g(0) + O(\xi h)$ and orthogonality condition (29) to get

$$y'_{n+1} - y'(t_{n+1}) = h^{s+1} g(0) \int_0^1 \prod_{i=1}^s (\xi - c_i) d\xi + O(h^{s+2}) = O(h^{s+2}). \quad (30)$$

The conclusion of the theorem follows from equations (25), (26), (30), and Definition 4.1. \square

Theorem 4.3 *If an s -stage FEPTRKN method has collocation parameters $(c_i)_{i=1}^s$ that satisfy*

$$\int_0^1 \xi^k \prod_{i=1}^s (\xi - c_i) d\xi = 0, \quad k = 0, 1, \quad (31)$$

then the method has step order $p = s + 2$.

Proof. From the relation $\mathbf{Y}_n - y(ch) = O(h^{s+2})$ (see Remark 4.1) and Taylor expansions of $f(t, y)$, one gets

$$f(ch, \mathbf{Y}_n) - f(ch, y(ch)) = O(h^{s+2}). \quad (32)$$

Let $R(t)$ be defined as in Theorem 4.1, i.e., $R(t) := u(t) - y(t)$. Then, we have $R(0) = 0$ and $R'(0) = 0$. It follows from (32) that

$$R''(c_i h) = u''(c_i h) - y''(c_i h) = f(c_i h, Y_{n,i}) - f(h, y(c_i h)) = O(h^{s+2}), \quad i = 1, \dots, s.$$

Thus, there exists a sufficiently smooth function $w(t)$ such that $w(c_i h) = 0$, $i = 1, \dots, s$, and

$$R''(t) = w(t) + O(h^{s+2}), \quad 0 \leq t \leq t_0 + T. \quad (33)$$

Again, there exists a sufficiently smooth function $g(t)$ such that $w(t) = g(t) \prod_{i=1}^s (t - c_i h)$. Therefore,

$$R''(t) = g(t) \prod_{i=1}^s (t - c_i h) + O(h^{s+2}), \quad 0 \leq t \leq t_0 + T. \quad (34)$$

Equation (34) and the relation $R'(0) = 0$ imply

$$R'(x) = \int_0^x R''(\xi) d\xi = \int_0^x \left(g(\xi) \prod_{i=1}^s (\xi - c_i h) + O(h^{s+2}) \right) d\xi. \quad (35)$$

Assigning $x := h$ in the above equality, we obtain

$$u'(h) - y'(h) = R'(h) = h^{s+1} \int_0^1 g(\xi h) \prod_{i=1}^s (\xi - c_i) d\xi + \int_0^h O(h^{s+2}) d\xi. \quad (36)$$

Using the Taylor expansion $g(\xi h) = g(0) + \xi h g'(0) + O(h^2)$ and the relation $\int_0^h O(h^{s+2}) d\xi = O(h^{s+3})$, we have

$$u'(h) - y'(h) = h^{s+1} g(0) \int_0^1 \prod_{i=1}^s (\xi - c_i) d\xi + h^{s+2} g'(0) \int_0^1 \xi \prod_{i=1}^s (\xi - c_i) d\xi + O(h^{s+3}).$$

This and orthogonality condition (31) imply

$$y'_{n+1} - y'(t_{n+1}) = u'(h) - y'(h) = O(h^{s+3}). \quad (37)$$

Similarly, from equation (34) and the relations $R(0) = 0$ and $R'(0) = 0$, one gets

$$R(x) = \int_0^x \int_0^t R''(\xi) d\xi dt = \int_0^x \int_0^t \left(g(\xi) \prod_{i=1}^s (\xi - c_i h) + O(h^{s+2}) \right) d\xi dt. \quad (38)$$

From equation (38), the relation $\int_0^h \int_0^t O(h^{s+2}) d\xi dt = O(h^{s+4})$, Fubini's Theorem, the Maclaurin expansion $g(\xi h) = g(0) + O(h)$, and orthogonality condition (31), one gets

$$\begin{aligned} y_{n+1} - y(t_{n+1}) &= R(h) = h^{s+2} \int_0^1 \int_0^t g(\xi h) \prod_{i=1}^s (\xi - c_i) d\xi dt + O(h^{s+4}) \\ &= h^{s+2} \int_0^1 \int_{\xi}^1 g(\xi h) \prod_{i=1}^s (\xi - c_i) dt d\xi + O(h^{s+4}) \\ &= h^{s+2} \int_0^1 g(\xi h) (1 - \xi) \prod_{i=1}^s (\xi - c_i) d\xi + O(h^{s+4}) \\ &= h^{s+2} g(0) \left(\int_0^1 \prod_{i=1}^s (\xi - c_i) d\xi - \int_0^1 \xi \prod_{i=1}^s (\xi - c_i) d\xi \right) + O(h^{s+3}) \\ &= O(h^{s+3}). \end{aligned} \quad (39)$$

From (37), (39), and Definition 4.1, one concludes that the method has step order $p = s + 2$.

Theorem 4.3 is proved. \square

Remark 4.2 From (24) with $w = 1 + c_i$, $i = 1, \dots, s$, and the relation $\int_0^{\omega h} \int_0^t O(h^{s+1}) d\xi dt = O(h^{s+3})$, we have

$$\begin{aligned} \mathbf{Y}_{n+1} - y(et_{n+1} + ch) &= u(\mathbf{e}h + ch) - y(\mathbf{e}h + ch) \\ &= h^{s+2} C_{n+1} \int_0^{e+c} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt + O(h^{s+3}), \quad n = 0, 1, \dots \end{aligned} \quad (40)$$

If \mathbf{Y}_0 is computed at a local accuracy order of $s+2$ at the initial step of computation, i.e., $\mathbf{Y}_0 - y(\mathbf{e}t_0 + ch) = O(h^{s+3})$, then we will have

$$\mathbf{Y}_n - y(et_n + ch) = h^{s+2} C_n \int_0^{e+c} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt + O(h^{s+3}), \quad n = 0, 1, \dots \quad (41)$$

Theorem 4.4 If an s -stage FEPTTRKN method has collocation parameters $(c_i)_{i=1}^s$ that satisfy

$$\int_0^1 \int_0^{1+x} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt dx = 0, \quad \int_0^1 \xi^k \prod_{i=1}^s (\xi - c_i) d\xi = 0, \quad k = 0, 1, 2, \quad (42)$$

then the method has step order $p = s + 3$.

Proof. Again, without loss of generality, we assume that $t_n = 0$. Let $R(t)$ be defined as in the proofs of Theorems 4.1 and 4.3, i.e., $R(t) = u(t) - y(t)$. Using Taylor expansions of $f(t, y)$ and (41), one gets

$$R''(ch) = u''(ch) - y''(ch) = f(ch, \mathbf{Y}_n) - f(ch, y(ch)) = h^{s+2} C \int_0^{e+c} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt + O(h^{s+3}),$$

where $C = C_n \frac{\partial f}{\partial y}(0, y(0))$. Thus, we have

$$R(0) = 0, \quad R'(0) = 0, \quad R''(c_i h) = C \int_0^{h+c_i h} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt + O(h^{s+3}), \quad i = 1, \dots, s.$$

Again, since $R''(t)$ is sufficiently smooth, there exists a smooth function $w(t)$ such that

$$R''(x) = w(x) + C \int_0^{h+x} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt + O(h^{s+3}), \quad w(c_i h) = 0, \quad i = 1, \dots, s. \quad (43)$$

Since $w(c_i h) = 0$, $i = 1, \dots, s$, there exists a sufficiently smooth function $g(x)$ such that (cf. Remark 4.1)

$$w(x) = g(x) \prod_{i=1}^s (x - c_i h).$$

This and (43) imply

$$R''(x) = g(x) \prod_{i=1}^s (x - c_i h) + C \int_0^{h+x} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt + O(h^{s+3}). \quad (44)$$

Integrate equation (44) and use the relation $R'(0) = 0$ to get

$$\begin{aligned}
R'(x) &= \int_0^x R''(\tau) d\tau \\
&= \int_0^x \left(g(\tau) \prod_{i=1}^s (\tau - c_i h) + C \int_0^{h+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt + O(h^{s+3}) \right) d\tau \\
&= \int_0^x g(\tau) \prod_{i=1}^s (\tau - c_i h) d\tau + C \int_0^x \int_0^{h+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt d\tau + \int_0^x O(h^{s+3}) d\tau.
\end{aligned}$$

Assigning $x := h$ in the above equality, we obtain

$$\begin{aligned}
u'(h) - y'(h) &= R'(h) = h^{s+1} \int_0^1 g(h\xi) \prod_{i=1}^s (\xi - c_i) d\xi \\
&\quad + Ch^{s+3} \int_0^1 \int_0^{1+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt d\tau + \int_0^h O(h^{s+3}) d\tau.
\end{aligned} \tag{45}$$

Using the Taylor expansion $g(\xi h) = \sum_{k=0}^2 \frac{g^{(k)}(0)}{k!} (\xi h)^k + O(h^3)$, we get

$$h^{s+1} \int_0^1 g(h\xi) \prod_{i=1}^s (\xi - c_i) d\xi = \sum_{k=0}^2 h^{s+1+k} \frac{g^{(k)}(0)}{k!} \int_0^1 \xi^k \prod_{i=1}^s (\xi - c_i) d\xi + O(h^{s+4}). \tag{46}$$

Equations (45)–(46) and the relation $\int_0^h O(h^{s+3}) d\tau = O(h^{s+4})$ imply

$$\begin{aligned}
u'(h) - y'(h) &= \sum_{k=0}^2 h^{s+1+k} \frac{g^{(k)}(0)}{k!} \int_0^1 \xi^k \prod_{i=1}^s (\xi - c_i) d\xi \\
&\quad + Ch^{s+3} \int_0^1 \int_0^{1+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt d\tau + O(h^{s+4}).
\end{aligned}$$

This and orthogonality conditions in (42) imply

$$y'_{n+1} - y(t_{n+1}) = u'(h) - y'(h) = O(h^{s+4}). \tag{47}$$

From equation (44) and the relations $R(0) = 0$ and $R'(0) = 0$, one gets

$$\begin{aligned}
R(x) &= \int_0^x \int_0^\zeta R''(\tau) d\tau d\zeta \\
&= \int_0^x \int_0^\zeta \left(g(\tau) \prod_{i=1}^s (\tau - c_i h) + C \int_0^{h+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt + O(h^{s+3}) \right) d\tau d\zeta.
\end{aligned} \tag{48}$$

This and the relation $\int_0^h \int_0^\zeta O(h^{s+3}) d\tau d\zeta = O(h^{s+5})$ imply

$$R(h) = \int_0^h \int_0^\zeta g(\tau) \prod_{i=1}^s (\tau - c_i h) d\tau d\zeta + C \int_0^h \int_0^\zeta \int_0^{h+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt d\tau d\zeta + O(h^{s+5}). \tag{49}$$

By substitutions, one gets

$$\int_0^h \int_0^\zeta \int_0^{h+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i h) d\xi dt d\tau d\zeta = h^{s+4} \int_0^1 \int_0^\zeta \int_0^{1+\tau} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt d\tau d\zeta = O(h^{s+4}). \tag{50}$$

From Fubini's Theorem one obtains

$$\begin{aligned} \int_0^h \int_0^\zeta g(\tau) \prod_{i=1}^s (\tau - c_i h) d\tau d\zeta &= \int_0^h \int_\tau^h g(\tau) \prod_{i=1}^s (\tau - c_i h) d\zeta d\tau \\ &= \int_0^h g(\tau) (h - \tau) \prod_{i=1}^s (\tau - c_i h) d\tau. \end{aligned} \quad (51)$$

Using the substitution $\tau = h\xi$ and the Maclaurin expansion $g(t) = \beta_0 + \beta_1 t + O(t^2)$, one gets

$$\begin{aligned} \int_0^h g(\tau) (h - \tau) \prod_{i=1}^s (\tau - c_i h) d\tau &= h^{s+2} \int_0^1 g(\xi h) (1 - \xi) \prod_{i=1}^s (\xi - c_i) d\xi \\ &= \sum_{k=0}^1 \beta_k h^{s+2+k} \int_0^1 \xi^k \prod_{i=1}^s (\xi - c_i) d\xi \\ &\quad - \sum_{k=0}^1 \beta_k h^{s+2+k} \int_0^1 \xi^{k+1} \prod_{i=1}^s (\xi - c_i) d\xi + O(h^{s+4}). \end{aligned} \quad (52)$$

From equations (51)–(52) and orthogonality conditions in (42), one obtains

$$\int_0^h \int_0^\zeta g(\tau) \prod_{i=1}^s (\tau - c_i h) d\tau d\zeta = O(h^{s+4}).$$

This and equations (49)–(50) imply

$$y_{n+1} - y(t_{n+1}) = u(h) - y(h) = R(h) = O(h^{s+4}). \quad (53)$$

It follows from (47), (53), and Definition 4.1 that the stage order of the method is $p = s + 3$.

Theorem (4.4) is proved. \square

Remark 4.3 From Fubini's Theorem, we have

$$\begin{aligned} \int_0^1 \int_0^{1+x} \int_0^t \prod_{i=1}^s (\xi - c_i) d\xi dt dx &= \int_0^1 \int_0^{1+x} \int_\xi^{1+x} \prod_{i=1}^s (\xi - c_i) dt d\xi dx \\ &= \int_0^1 \int_0^{1+x} (1 + x - \xi) \prod_{i=1}^s (\xi - c_i) d\xi dx. \end{aligned}$$

If (42) holds, then by Fubini's Theorem one gets

$$\begin{aligned} \int_0^1 \int_0^1 (1 + x - \xi) \prod_{i=1}^s (\xi - c_i) d\xi dx &= \int_0^1 \int_0^1 (1 + x - \xi) \prod_{i=1}^s (\xi - c_i) dx d\xi \\ &= \int_0^1 \left(\frac{3}{2} - \xi \right) \prod_{i=1}^s (\xi - c_i) d\xi \\ &= \frac{3}{2} \int_0^1 \prod_{i=1}^s (\xi - c_i) d\xi - \int_0^1 \xi \prod_{i=1}^s (\xi - c_i) d\xi = 0. \end{aligned}$$

Moreover, using Fubini's Theorem, one gets

$$\begin{aligned} \int_0^1 \int_1^{1+x} (1+x-\xi) \prod_{i=1}^s (\xi - c_i) d\xi dx &= \int_1^2 \int_{\xi-1}^1 (1+x-\xi) \prod_{i=1}^s (\xi - c_i) dx d\xi \\ &= \int_1^2 \frac{(\xi-2)^2}{2} \prod_{i=1}^s (\xi - c_i) d\xi. \end{aligned}$$

Therefore, condition (42) is equivalent to

$$\int_1^2 (\xi-2)^2 \prod_{i=1}^s (\xi - c_i) d\xi = 0, \quad \int_0^1 \xi^k \prod_{i=1}^s (\xi - c_i) d\xi = 0, \quad k = 0, 1, 2. \quad (54)$$

In practice, we use (54) instead of (42) to find collocation parameters $(c_i)_{i=1}^s$.

4.3 Stability

Applying an FEPTKRN method given by $(\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{d})$ to the Dahlquist test equation

$$y'' = \lambda y, \quad y(0) = y_0, \quad y'(0) = y'_0,$$

we get the iterations

$$\begin{aligned} \mathbf{Y}_{n+1} &= \mathbf{e} y_{n+1} + \mathbf{c} h y'_{n+1} + \lambda h^2 \mathbf{A}(t_n, h) \mathbf{Y}_n, \\ y_{n+1} &= y_n + h y'_n + \lambda h^2 \mathbf{b}^T(t_n, h) \mathbf{Y}_n, \\ y'_{n+1} &= y'_n + \lambda h \mathbf{d}^T(t_n, h) \mathbf{Y}_n. \end{aligned} \quad (55)$$

This implies

$$\begin{pmatrix} \mathbf{Y}_{n+1} \\ y_{n+1} \\ h y'_{n+1} \end{pmatrix} = \mathbf{M}_{(t_n, h)}(z) \begin{pmatrix} \mathbf{Y}_n \\ y_n \\ h y'_n \end{pmatrix}, \quad z = \lambda h^2, \quad (56)$$

where

$$\mathbf{M}_{(t_n, h)}(z) = \begin{pmatrix} z[\mathbf{A}(t_n, h) + \mathbf{e} \mathbf{b}^T(t_n, h) + \mathbf{c} \mathbf{d}^T(t_n, h)] & \mathbf{e} & \mathbf{e} + \mathbf{c} \\ z \mathbf{b}^T(t_n, h) & 1 & 1 \\ z \mathbf{d}^T(t_n, h) & 0 & 1 \end{pmatrix}.$$

Observe that the formula of our amplification matrix $\mathbf{M}_{(t_n, h)}(z)$ differs from that in [3, 6], where they instead use the formulation

$$\begin{pmatrix} \mathbf{Y}_n \\ y_{n+1} \\ h y'_{n+1} \end{pmatrix} = \tilde{\mathbf{M}}(z) \begin{pmatrix} \mathbf{Y}_{n-1} \\ y_n \\ h y'_n \end{pmatrix}, \quad \tilde{\mathbf{M}}(z) = \begin{pmatrix} z \mathbf{A} & \mathbf{e} & \mathbf{c} \\ z^2 \mathbf{b}^T \mathbf{A} & 1 + z \mathbf{b}^T \mathbf{e} & 1 + z \mathbf{b}^T \mathbf{c} \\ z^2 \mathbf{d}^T \mathbf{A} & z \mathbf{d}^T \mathbf{e} & 1 + z \mathbf{d}^T \mathbf{c} \end{pmatrix}, \quad z = \lambda h^2. \quad (57)$$

Note that in [3, 6] the coefficients $\mathbf{A}(t, h)$, $\mathbf{b}(t, h)$ and $\mathbf{d}(t, h)$ are independent of t and h . From (55), we have

$$\begin{pmatrix} \mathbf{Y}_{n+1} \\ y_{n+1} \\ h y'_{n+1} \end{pmatrix} = \begin{pmatrix} z \mathbf{A} & \mathbf{e} & \mathbf{c} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{Y}_n \\ y_{n+1} \\ h y'_{n+1} \end{pmatrix}.$$

This and equation (57) imply

$$\begin{pmatrix} \mathbf{Y}_{n+1} \\ y_{n+1} \\ h y'_{n+1} \end{pmatrix} = \begin{pmatrix} z \mathbf{A} & \mathbf{e} & \mathbf{c} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tilde{\mathbf{M}}(z) \begin{pmatrix} z \mathbf{A} & \mathbf{e} & \mathbf{c} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{Y}_n \\ y_n \\ h y'_n \end{pmatrix}. \quad (58)$$

From (56) and (58), we have the similarity transformation

$$\mathbf{M}(z) = \begin{pmatrix} z\mathbf{A} & \mathbf{e} & \mathbf{c} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tilde{\mathbf{M}}(z) \begin{pmatrix} z\mathbf{A} & \mathbf{e} & \mathbf{c} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1}.$$

This means that although the two formulations lead to different amplification matrices, the spectral radii of these matrices are the same. Note that for conventional EPTRKN methods, we have $\mathbf{d}^T \mathbf{e} = 1$ and $\mathbf{b}^T \mathbf{e} = \frac{1}{2}$ and, therefore, the matrix $\tilde{\mathbf{M}}(z)$ can be simplified further. However, for general basis functions, the coefficients \mathbf{A} , \mathbf{b} , and \mathbf{d} of a FEPTRKN method depend on both t and h and we do not have $\mathbf{d}^T \mathbf{e} = 1$ and $\mathbf{b}^T \mathbf{e} = \frac{1}{2}$.

The coefficients $\mathbf{b}(t_n, h)$, $\mathbf{d}(t_n, h)$, and $\mathbf{A}(t_n, h)$ are independent of t_n for the so-called class of *separable* basis functions $\{u_i\}_{i=1}^s$, i.e., if we have (see [15, Section 3])

$$\mathbf{u}(x + y) = \mathcal{F}(y)\mathbf{u}(x), \quad \forall x, y \in \mathbb{R}$$

where $\mathbf{u} = (1, t, u_1, \dots, u_s)^T$ and \mathcal{F} is a square matrix of size $(s + 2) \times (s + 2)$ with entries that are univariate functions. This class includes the most common functions such as polynomial, exponential, and trigonometric functions. In this particular class, the stability region of the method for a given stepsize h can be defined as

$$S_h := \{z \in (-\infty, 0] : \rho(\mathbf{M}_h(z)) \leq 1\} \quad (59)$$

where $\rho(\mathbf{M}_h(z))$ denotes the spectral radius of $\mathbf{M}_h(z)$ and

$$\mathbf{M}_{(h)}(z) = \begin{pmatrix} z[\mathbf{A}(h) + \mathbf{e}\mathbf{b}^T(h) + \mathbf{c}\mathbf{d}^T(h)] & \mathbf{e} & \mathbf{e} + \mathbf{c} \\ z\mathbf{b}^T(h) & 1 & 1 \\ z\mathbf{d}^T(h) & 0 & 1 \end{pmatrix}. \quad (60)$$

It becomes possible to characterize the stability region numerically or even analytically for some special cases (cf. [11], [13], [14]). Furthermore, if we expand the coefficients into Taylor series

$$\begin{aligned} a_{ij} &= a_{ij}^{(0)} + a_{ij}^{(1)}h + a_{ij}^{(2)}h^2 + \dots, \\ b_i &= b_i^{(0)} + b_i^{(1)}h + b_i^{(2)}h^2 + \dots, \end{aligned}$$

then it can be shown that the leading terms are constant and conform to the coefficients of the conventional EPTRKN method defined by \mathbf{c} (cf. [16]). Thus, the coefficients $\mathbf{b}(t, h)$, $\mathbf{d}(t, h)$, $\mathbf{A}(t, h)$ converge to those of the conventional EPTRKN method when $h \rightarrow 0$. Consequently, the amplification matrix $\mathbf{M}_{(t_n, h)}(z)$ and the stability region of a FEPTRKN method converge to the amplification matrix $\mathbf{M}(z)$ and the stability region of the corresponding conventional EPTRKN method, respectively.

5 Extensions

We now discuss some aspects that are essential to producing competitive numerical codes.

5.1 Variable stepsize

When the stepsize h_n is accepted and the next stepsize h_{n+1} is suggested, the values in the next step are computed by

$$\begin{aligned} \mathbf{Y}_{n+1} &= \mathbf{e}y_{n+1} + h_{n+1}\mathbf{c}y'_{n+1} + h_{n+1}^2\mathbf{A}(t_n, h_n, h_{n+1})f(\mathbf{e}t_n + \mathbf{c}h_n, \mathbf{Y}_n), \\ y_{n+2} &= y_{n+1} + h_{n+1}y'_{n+1} + h_{n+1}^2\mathbf{b}^T(t_{n+1}, h_{n+1})f(\mathbf{e}t_{n+1} + \mathbf{c}h_{n+1}, \mathbf{Y}_{n+1}), \\ y'_{n+2} &= y'_{n+1} + h_{n+1}\mathbf{d}^T(t_{n+1}, h_{n+1})f(\mathbf{e}t_{n+1} + \mathbf{c}h_{n+1}, \mathbf{Y}_{n+1}). \end{aligned} \quad (61)$$

The coefficients $\mathbf{b}^T(t_{n+1}, h_{n+1})$ and $\mathbf{d}^T(t_{n+1}, h_{n+1})$ are computed from the first two equations in (9) with $t = t_{n+1}$ and $h = h_{n+1}$. However, we cannot solve for $\mathbf{A}(t_n, h_n, h_{n+1})$ from the third equation in (9) if $h_{n+1} \neq h_n$. In particular, the coefficients $\mathbf{A}(t_n, h_n, h_{n+1})$ are obtained by solving the systems

$$\left(\mathbf{v}_1(t_n, h_n, h_{n+1}), \dots, \mathbf{v}_s(t_n, h_n, h_{n+1}) \right) = h_{n+1}^2 \mathbf{A}(t_n, h_n, h_{n+1}) \mathbf{F}(t_n, h_n),$$

where

$$\mathbf{v}_i(t_n, h_n, h_{n+1}) = u_i(et_{n+1} + ch_{n+1}) - u_i(et_{n+1}) - ch_{n+1}u'_i(t_{n+1}), \quad i = 1, \dots, s.$$

Note that even a conventional variable stepsize EPTRKN induces variable coefficients. However, \mathbf{b} and \mathbf{d} remain constant and the update for \mathbf{A} leads to a simple diagonal scaling [6]. The next approximations $y_{n+2} \approx y(t_{n+1} + h_{n+1})$ and $y'_{n+2} \approx y'(t_{n+1} + h_{n+1})$ resulting from the new stepsize h_{n+1} are subject to a local truncation error denoted by LTE which is often estimated by using another embedded method (see Section 5.3 below). If the estimated error LTE is smaller than a prescribed tolerance TOL, then h_{n+1} is accepted. Otherwise, it is rejected and a reduced stepsize \tilde{h}_{n+1} is used to recompute $\mathbf{Y}_{n+1} \approx y(et_{n+1} + c\tilde{h}_{n+1})$, $y'_{n+2} \approx y'(t_{n+1} + \tilde{h}_{n+1})$, and $y_{n+2} \approx y(t_{n+1} + \tilde{h}_{n+1})$. This process is repeated until an accepted value of h_{n+1} is found.

Varying the stepsize from a collocation perspective means that, from the past accepted values y_n, y'_n , and \mathbf{Y}_n we construct the collocation solution $u(t)$ defined in (10) and then evaluate and store the values $y_{n+1} = u(t_n + h_n)$, $y'_{n+1} = u'(t_n + h_n)$. The next stage values \mathbf{Y}_{n+1} are generated from this collocation function using an estimated stepsize h_{n+1} (see Section 5.4 below). The acceptance of this h_{n+1} is subject to a local truncation error for computing y_{n+2} using this stepsize (cf. Section 5.4). Therefore, the collocation function $u(t)$ and values y_{n+1}, y'_{n+1} remain the same when h_{n+1} varies. We only adjust the stepsize h_{n+1} for computing \mathbf{Y}_{n+1} , specifically, $\mathbf{Y}_{n+1} = u(et_{n+1} + ch_{n+1})$, see (11). As a consequence, the following generalization of [6, Theorem 2.1] holds.

Theorem 5.1 *The s -stage variable stepsize FEPTRKN method (61) is of stage order $p = s$ and of step order at least $p = s$ for any set of distinct collocation points $(c_i)_{i=1}^s$. It has step order $p = s + q$, $q = 1, 2$, if the $(c_i)_{i=1}^s$ satisfy the orthogonality conditions*

$$\int_0^1 \xi^k \prod_{i=1}^s (\xi - c_i) d\xi = 0, \quad k = 0, q - 1.$$

5.2 Interpolation

A continuous extension of an s -stage FEPTRKN method determined by $(\mathbf{c}, \mathbf{A}(t_n, h_n), \mathbf{b}(t_n, h_n), \mathbf{d}(t_n, h_n))$ is defined as follows (cf. [6], [14])

$$y_{n+\xi} = y_n + \xi h_n y'_n + (\xi h_n)^2 \mathbf{b}^T(t_n, h_n, \xi) f(et_n + ch, \mathbf{Y}_n), \quad (62)$$

$$y'_{n+\xi} = y'_n + \xi h_n \mathbf{d}^T(t_n, h_n, \xi) f(et_n + ch, \mathbf{Y}_n), \quad 0 \leq \xi \leq 1, \quad (63)$$

where $y_{n+\xi} \approx y(t_{n+\xi})$ and $y'_{n+\xi} \approx y'(t_{n+\xi})$ with $t_{n+\xi} := t_n + \xi h_n$, and the coefficients $\mathbf{b}(t_n, h_n, \xi)$ and $\mathbf{d}(t_n, h_n, \xi)$ are obtained from the equations

$$\begin{aligned} \left(u_1(t_{n+\xi}) - u_1(t_n) - \xi h_n u'_1(t_n), \dots, u_s(t_{n+\xi}) - u_s(t_n) - \xi h_n u'_s(t_n) \right) &= (\xi h_n)^2 \mathbf{b}^T(t_n, h_n, \xi) \mathbf{F}(t_n, h_n), \\ \left(u_1(t_{n+\xi}) - u_1(t_n), \dots, u_s(t_{n+\xi}) - u_s(t_n) \right) &= \xi h_n \mathbf{d}^T(t_n, h_n, \xi) \mathbf{F}(t_n, h_n). \end{aligned} \quad (64)$$

In other words, any $y_{n+\xi}$ and $y'_{n+\xi}$ are retrieved as if $t_{n+\xi} = t_n + \xi h_n$ was the end point. From their definitions one has $y_{n+\xi} = u(t_n + \xi h_n)$ and $y'_{n+\xi} = u'(t_n + \xi h_n)$, where $u(t)$ is the collocation solution defined in (10). From (24) it is easy to check that $y_{n+\xi} - y(t_n + \xi h_n) = O(h_n^{s+2})$ and $y'_{n+\xi} - y'(t_n + \xi h_n) = O(h_n^{s+1})$, $\forall \xi \in [0, 1]$. Hence, the following result holds.

Theorem 5.2 *The FEPTTRKN method defined by (62)–(63) with \mathbf{b} and \mathbf{d} defined by (64) produces a continuous FEPTTRKN method of order s , i.e.,*

$$y(t_n + \xi h_n) - y_{n+\xi} = O(h_n^{s+2}), \quad y'(t_n + \xi h_n) - y'_{n+\xi} = O(h_n^{s+1}), \quad 0 \leq \xi \leq 1.$$

It can further be shown that when the collocation functions are the monomials, the coefficient $\mathbf{b}(t_n, h_n, \xi)$ depends only on ξ (cf. [6]).

5.3 Embedded methods

Consider an s -stage FEPTTRKN method with coefficients $(\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{d})$. We now wish to have an embedded FEPTTRKN method $(\tilde{\mathbf{c}}, \tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{d}})$ paired with the FEPTTRKN method $(\mathbf{c}, \mathbf{A}, \mathbf{b}, \mathbf{d})$ to cheaply estimate the errors and control the stepsizes in practice.

Let $\tilde{s} < s$, $\{\tilde{c}_1, \dots, \tilde{c}_{\tilde{s}}\} \subsetneq \{c_1, \dots, c_s\}$, $\tilde{\mathbf{c}} = (\tilde{c}_1, \dots, \tilde{c}_{\tilde{s}})^T$ and $\tilde{\mathbf{e}} = (1, \dots, 1)^T$ of length \tilde{s} . An embedded pair FEPTTRKN methods in which another approximation \tilde{y}_{n+1} to $y(t_{n+1})$ can be computed without extra function evaluations is defined by

$$\begin{aligned} y_{n+1} &= y_n + h_n y'_n + h_n^2 \mathbf{b}^T(t_n, h_n) f(\mathbf{e}t_n + \mathbf{c}h_n, \mathbf{Y}_n), \\ y'_{n+1} &= y'_n + h_n \mathbf{d}^T(t_n, h_n) f(\mathbf{e}t_n + \mathbf{c}h_n, \mathbf{Y}_n), \\ \tilde{y}_{n+1} &= y_n + h_n y'_n + h_n^2 \tilde{\mathbf{b}}^T(t_n, h_n) f(\tilde{\mathbf{e}}t_n + \tilde{\mathbf{c}}h_n, \tilde{\mathbf{Y}}_n), \\ \mathbf{Y}_{n+1} &= \mathbf{e}y_{n+1} + \mathbf{c}h_n y'_{n+1} + h_n^2 \mathbf{A}(t_n, h_n) f(\mathbf{e}t_n + \mathbf{c}h_n, \mathbf{Y}_n) \end{aligned} \tag{65}$$

where $\tilde{\mathbf{Y}}_n = (\tilde{Y}_{n,1}, \dots, \tilde{Y}_{n,\tilde{s}})$ and $f(\tilde{\mathbf{e}}t_n + \tilde{\mathbf{c}}h_n, \tilde{\mathbf{Y}}_n) = (f(t_n + \tilde{c}_1 h_n, \tilde{Y}_{n,1}), \dots, f(t_n + \tilde{c}_{\tilde{s}} h_n, \tilde{Y}_{n,\tilde{s}}))^T$ are defined according to the mapping (which is well defined because the c_i are distinct)

$$\text{if } \tilde{c}_i = c_j \quad \text{then} \quad \tilde{Y}_{n,i} = Y_{n,j}.$$

The coefficient $\tilde{\mathbf{b}}(t_n, h_n)$ is obtained by solving the linear system

$$\left(\tilde{u}_1(t_n + h_n) - \tilde{u}_1(t_n) - h_n \tilde{u}'_1(t_n), \dots, \tilde{u}_{\tilde{s}}(t_n + h_n) - \tilde{u}_{\tilde{s}}(t_n) - h_n \tilde{u}'_{\tilde{s}}(t_n) \right) = h_n^2 \tilde{\mathbf{b}}^T(t_n, h_n) \tilde{\mathbf{F}}(t_n, h_n)$$

where

$$\tilde{\mathbf{F}}(t_n, h_n) = \left(\tilde{u}_1''(\tilde{\mathbf{e}}t_n + \tilde{\mathbf{c}}h_n), \dots, \tilde{u}_{\tilde{s}}''(\tilde{\mathbf{e}}t_n + \tilde{\mathbf{c}}h_n) \right).$$

In other words, $\tilde{\mathbf{b}}$ is the coefficient of the FEPTTRKN method generated from $\tilde{\mathbf{c}}$ and the basis of functions $\{\tilde{u}_1, \dots, \tilde{u}_{\tilde{s}}\} \subsetneq \{u_1, \dots, u_s\}$. The solution \tilde{y}_{n+1} is computed by combining the subset of past stage values with indices corresponding to $\tilde{\mathbf{c}}$. For this definition of $\tilde{\mathbf{b}}$ we can assert that $\tilde{y}_{n+1} = y(t_{n+1}) + O(h^{\tilde{s}+1})$ as a result of Theorem 4.1.

Theorem 5.3 *An s -stage embedded pair FEPTTRKN (65) produces solutions y_{n+1} and \tilde{y}_{n+1} that satisfy*

$$y_{n+1} - \tilde{y}_{n+1} = O(h^{\tilde{s}+1}),$$

for all set of distinct collocation parameters $(c_i)_{i=1}^s$.

5.4 Error control and step-size change

Let us discuss a strategy for changing stepsizes in the implementation of a FEPTKRN method of order p embedded with a FEPTKRN method of order $\tilde{p} < p$ using the variable stepsize technique in Section 5.1. At each step we compute a local truncation error LTE as follows

$$\text{LTE} = \|y_{n+1} - \tilde{y}_{n+1}\| = O(h^{\tilde{p}+1}). \quad (66)$$

Formula (66) for computing LTE is simpler than the one in [6] where LTE is computed by

$$\text{LTE} = \sqrt{\|y_{n+1} - \tilde{y}_{n+1}\|^2 + \|y'_{n+1} - \tilde{y}'_{n+1}\|^2} = O(h^{\tilde{p}+1}). \quad (67)$$

Here \tilde{y}'_{n+1} is computed by

$$\tilde{y}'_{n+1} = y'_n + h_n \tilde{\mathbf{d}}^T(t_n, h_n) f(\tilde{\mathbf{c}}t_n + \tilde{\mathbf{c}}h_n, \tilde{\mathbf{Y}}_n),$$

where $\tilde{\mathbf{Y}}_n$ is defined as in Section 5.3 and $(\tilde{\mathbf{c}}, \tilde{\mathbf{A}}, \tilde{\mathbf{b}}, \tilde{\mathbf{d}})$ are the coefficients of the embedded method.

In our experiments we have found out that using (67) instead of (66) for computing LTE results in having smaller stepsizes, and, therefore, yields smaller errors. However, having smaller stepsizes makes the FEPTKRN methods use more function evaluations. From numerical experiments, we do not see any advantage of using (67) over using (66) for computing LTE.

A stepsize h_n is accepted if $\text{LTE} \leq \text{TOL}$ and rejected if otherwise. If h_n is rejected, then we repeat the computations with the new stepsize $h_n = h_n/2$ until an accepted h_n is found. If h_n is accepted, then the stepsize h_{n+1} in the next step is chosen as

$$h_{n+1} = h_n \min \left\{ 2, \max \left\{ 0.5, 0.8(\text{TOL}/\text{LTE})^{1/(\tilde{p}+1)} \right\} \right\}.$$

It follows from this formula that the ratio $\frac{h_{n+1}}{h_n}$ always stays in the interval $[0.5, 2]$. This stepsize changing technique was also used in [6].

6 Numerical experiments

The experiments were conducted in double precision (machine precision = 0.2×10^{-15}) using MATLAB on a PC computer with 2Gb RAM and 2.8 GHz.

6.1 Derivation of some methods

We derived some methods with the following bases:

1. $\{x^2, x^3, x^4\}$ for eptrkn52.
2. $\{x^2, \cos(\omega x), \sin(\omega x)\}$ for feptrkn52.
3. $\{x^2, x^3, x^4, x^5\}$ for eptrkn73.
4. $\{\cos(\omega x), \sin(\omega x), \cos(2\omega x), \sin(2\omega x)\}$ for feptrkn73.
5. $\{x^2, x^3, x^4, x^5, x^6\}$ for eptrkn84.
6. $\{x^2, \cos(\omega x), \sin(\omega x), \cos(2\omega x), \sin(2\omega x)\}$ for feptrkn84.
7. $\{x^2, x^3, x^4, x^5, x^6, x^7\}$ for eptrkn95.

8. $\{\cos(\omega x), \sin(\omega x), \cos(2\omega x), \sin(2\omega x), \cos(3\omega x), \sin(3\omega x)\}$ for feptrkn95.

The coefficients of FEPTRKN methods based on bases 2, 4, 6, and 8 above are independent of t but they are functions of ωh . The coefficients of EPTRKN methods, i.e., methods based on bases 1, 3, 5, and 7 above, are constants.

Methods based on bases 1 and 2 are 3-stage methods and are implemented with

$$\mathbf{c} = [0.18677613705141 \quad 0.75202972313575 \quad 1.66119413981284]^T.$$

This set of collocation points $(c_i)_{i=1}^3$ are determined to satisfy orthogonal conditions in (31) and the equation

$$\int_0^2 \prod_{i=1}^3 (\xi - c_i) d\xi = 0.$$

These methods have the same step order $p = 5$ by Theorem 4.3 and are denoted by eptrkn52 for basis 1 and feptrkn52 for basis 2. Their embedded methods have the same step order $p = 2$.

Methods based on bases 3 and 4 are 4-stage methods. We implement these methods with

$$\mathbf{c} = [10027252023777 \quad 0.46050359576754 \quad 0.86389485661306 \quad 1.43247188452449]^T$$

which is computed to satisfy orthogonality conditions in (42). These methods share the same accuracy order $p = 8$ by Theorem 4.4 and are denoted by eptrkn73 and feptrkn73 for basis 3 and basis 4, respectively. Their embedded methods are 3-stage methods and have the same accuracy order $p = 3$.

Methods based on bases 5 and 6 are 5-stage methods. We implement these methods with

$$\mathbf{c} = [0.0911311145011 \quad 0.4288524464674 \quad 0.8402456535427 \quad 1.3131095250315 \quad 1.8405501493461]^T$$

which is computed to satisfy orthogonality conditions in (42) and the equation

$$\int_0^2 \prod_{i=1}^5 (\xi - c_i) d\xi = 0.$$

These methods have the same accuracy order $p = 8$ by Theorem 4.4 and are denoted by eptrkn84 for basis 5 and feptrkn84 for basis 6. Their embedded methods are 4-stage methods and have the same accuracy order $p = 4$.

Methods based on bases 7 and 8 are 6-stage methods. These methods are implemented with

$$\mathbf{c} = [0 \quad 0.15981788694649 \quad 0.47315766336506 \quad 0.80767247891979 \quad 1 \quad 1.55935197076839]^T$$

which is computed so that orthogonality conditions in (42) hold and that the set $(c_i)_{i=1}^6$ contains two values 0 and 1. Having 0 and 1 in $(c_i)_{i=1}^6$ helps reducing one right-hand side function evaluation per step by reusing $f(t_{n+1}, y_{n+1})$ computed from the previous step. These methods share the same accuracy order $p = 9$ by Theorem 4.4 and are denoted by eptrkn95 for basis 7 and feptrkn95 for basis 8. Their embedded methods are 5-stage methods and have the same accuracy order $p = 5$.

6.2 Stability regions

As mentioned earlier, the coefficients of FEPTRKN methods converge to those of the corresponding EPTRKN methods when h goes to zero. Thus, the stability regions of FEPTRKN methods converge to those of the corresponding EPTRKN methods when h approaches to zero.

Using equations (59)–(60) we plot the stability regions of the feptrkn52, feptrkn73, feptrkn84, and feptrkn95 methods with constant stepsizes, i.e., $h_n = h = \text{const}$. Figure 1 plots the stability regions

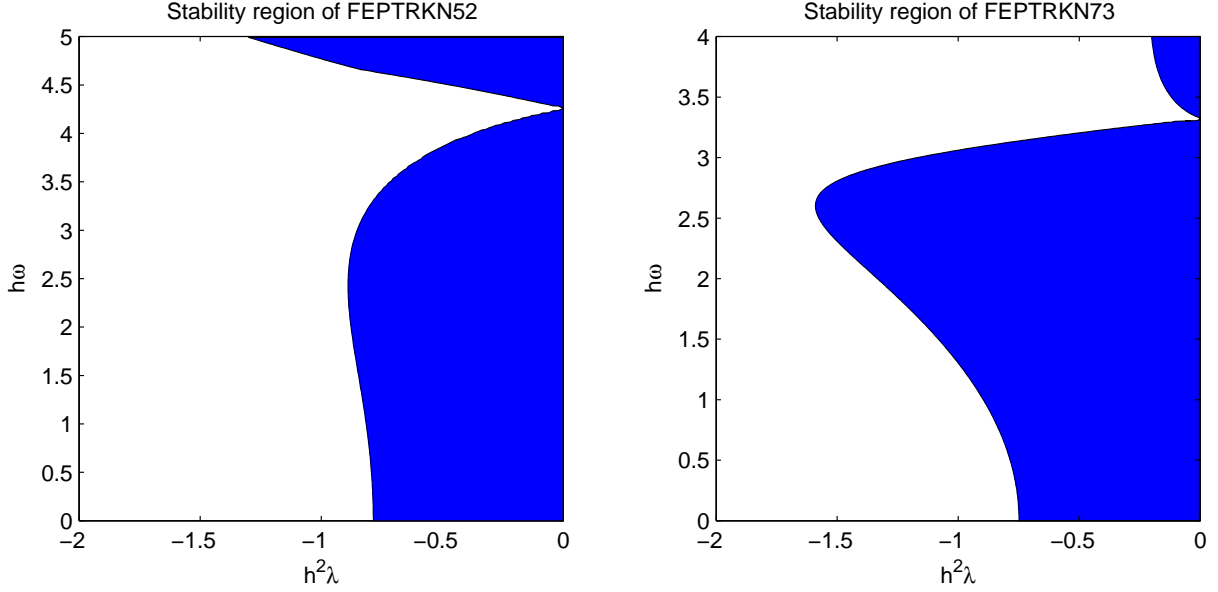


Figure 1: Plots of stability regions of the feptrkn52 and feptrkn73 methods.

of the feptrkn52 method for $\omega h \in [0, 5]$ (left) and of the feptrkn73 method for $\omega h \in [0, 4]$ (right). It can be seen from the left figure in Figure 1 that the stability region of the feptrkn52 method is greater than the stability region of the eptrkn52 method for $\omega h \in [0, 3]$. From the right figure in Figure 1 one concludes that the stability region of the feptrkn74 method is larger than that of the eptrkn74 method when $\omega h \in (0, 2.7]$. To have large stability regions, one should not use $\omega h > 3.5$ and $\omega h > 3$ for the feptrkn52 and feptrkn73 methods, respectively.

Figure 2 plots the stability regions of the feptrkn84 method (left) and the feptrkn95 method (right) for $\nu = \omega h \in [0, 4]$. It can be seen from the left figure in Figure 2 that the stability region of the feptrkn84 method for $\omega h \in [0, 3.4]$ is larger than that of the eptrkn84 method. The right figure in Figure 2 indicates that the stability region of the feptrkn95 method when $\omega h \in [0, 2.5]$ is larger than that of the eptrkn95 method. Figure 2 suggests that to have large stability regions one should not use $\omega h > 3.5$ and $\omega h > 2.8$ for the feptrkn84 method and the feptrkn95 method, respectively.

Although the stability regions of (F)EPTRKN methods are not as large as those of Runge-Kutta-Nyström methods, they are still sufficiently large for solving nonstiff problems (cf. [3], [6]).

6.3 Test problems

Numerical experiments are carried out with the following problems:

BETT-Consider the system (see [17, p.141])

$$\begin{aligned} y_1'' &= -y_1 + 0.001 \cos t, & y_1(0) &= 1, & y_1'(0) &= 0, \\ y_2'' &= -y_2 + 0.001 \sin t, & y_2(0) &= 0, & y_2'(0) &= 0.9995, \end{aligned}$$

where the integration domain is $[0, 40]$. The solution to the problem is given by

$$\begin{aligned} y_1(t) &= \cos t + 0.0005t \sin t, \\ y_2(t) &= \sin t - 0.0005t \cos t. \end{aligned}$$

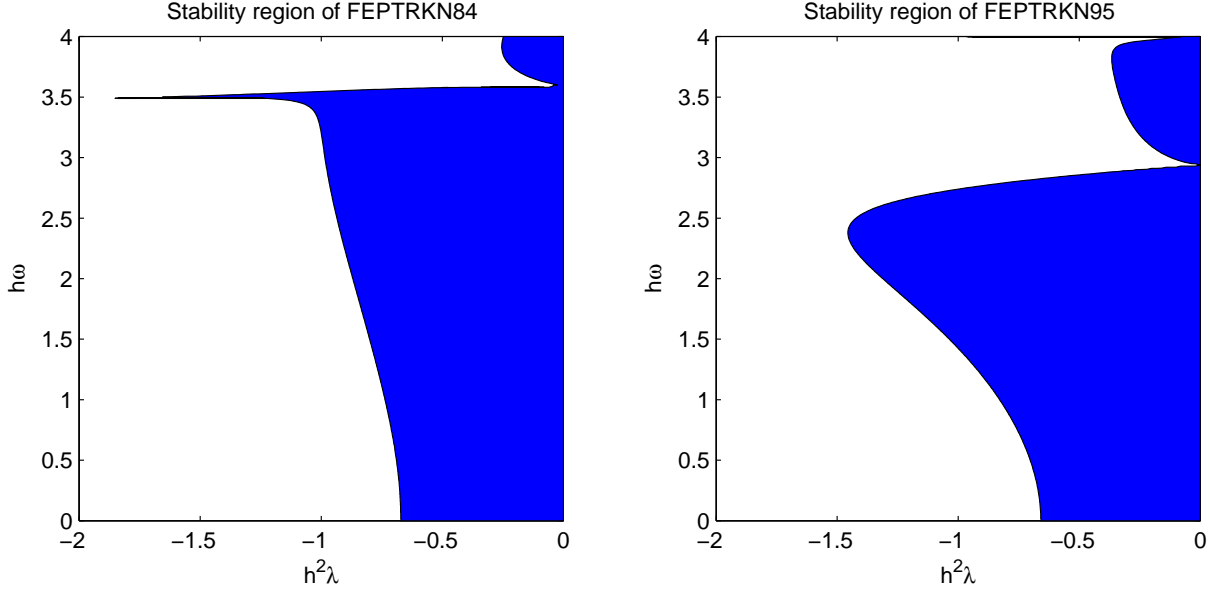


Figure 2: Plots of stability regions of the feptrkn84 and feptrkn95 methods.

NEWT-The two-body gravitational problem

$$\begin{aligned} y_1'' &= -\frac{y_1}{(y_1^2 + y_2^2)^{3/2}}, & y_1(0) &= 1 - e, & y_1'(0) &= 0, \\ y_2'' &= -\frac{y_2}{(y_1^2 + y_2^2)^{3/2}}, & y_2(0) &= 0, & y_2'(0) &= \sqrt{\frac{1+e}{1-e}}, \end{aligned}$$

whose exact solution is given by

$$\begin{aligned} y_1(t) &= \cos u - e, \\ y_2(t) &= \sqrt{1 - e^2} \sin u, \end{aligned}$$

where u is the solution of Kepler's equation $u = t + e \sin u$. The integration domain for this problem is $[0, 20]$.

6.4 Results and discussion

Table 1 presents numerical results for solving the BETT problem using the four methods eptrkn52, eptrkn73, eptrkn84, and eptrkn95 implemented with constant stepsizes. The NCD values in Table 1 are computed by the formula

$$\text{NCD} = \log_{10} \max(E_1, E_2), \quad E_i := \max_{0 \leq t_n \leq T} |y_i^{\text{comput}}(t_n) - y_i(t_n)|.$$

The NCD values can be used to determine accuracy order of a method in practice. In particular, if a method has an accuracy order p , then the error for computing $y_i(t_n)$ is of the form Ch^p . Thus, the NCD values for computing $y_i(t_n)$ is about $\log_{10} C + p \log_{10} h$. Therefore, the difference between the NCD values at stepsize h and at stepsize $h/2$ is about $p \log_{10} 2 \approx 0.3p$. Thus, one can estimate p the accuracy order

of the method by finding differences between NCD values at stepsize h and $h/2$ and then dividing these differences by 0.3.

By subtracting the NCD values at stepsize $h/2$ from the NCD values at stepsize h and then dividing the results by 0.3 one can see that the accuracy orders of the eptrkn52, eptrkn73, eptrkn84, and eptrkn95 methods are $p = 5$, $p = 7$, $p = 8$, and $p = 9$, respectively. This agrees with the super-convergence results in Theorems 4.3 and 4.4. We also see from Table 1 that the higher accuracy order of the methods, the more accurate they are for the same stepsizes h . However, higher order methods use more function evaluations in each step. Thus, the data in Table 1 does not tell us which method is the best in this experiment. The main purpose of using the results in Table 1 is to verify the super-convergence results in Theorems 4.3 and 4.4.

Table 1: NCD values for the BETT problem

h	eptrkn52	eptrkn73	eptrkn84	eptrkn95
$1/2^1$	-2.6	-4.0	-6.0	-5.9
$1/2^2$	-4.1	-6.3	-8.2	-8.7
$1/2^3$	-5.7	-8.7	-10.8	-11.7
$1/2^4$	-7.2	-11.1	-13.5	-14.6
$1/2^5$	-8.7	-13.5	-15.1	-14.3
$1/2^6$	-10.2	-15.5	-15.7	-14.5
$1/2^7$	-11.7	-14.7	-14.4	-14.7
$1/2^8$	-13.2	-14.3	-14.3	-14.4
$1/2^9$	-14.5	-14.6	-14.5	-14.9

Table 2 presents numerical results for solving the NEWT problem with the four EPTRKN methods eptrkn52, eptrkn73, eptrkn84, and eptrkn95. Again by subtracting the NCD values at stepsize $h/2$ from the NCD values at stepsize h and then dividing the obtained results by 0.3, we can see that the accuracy orders of the eptrkn52, eptrkn73, eptrkn84, and eptrkn95 methods are $p = 5$, $p = 7$, $p = 8$, and $p = 9$, respectively.

Table 2: NCD values for the NEWT problem, $e = 0.01$.

h	eptrkn52	eptrkn73	eptrkn84	eptrkn95
$1/2$	-0.9	-2.2	-2.6	-2.9
$1/2^2$	-2.4	-4.5	-6.2	-6.0
$1/2^3$	-3.9	-6.9	-8.9	-9.2
$1/2^4$	-5.4	-9.2	-11.5	-12.1
$1/2^5$	-6.9	-11.5	-13.6	-13.7
$1/2^6$	-8.4	-12.6	-13.7	-13.3
$1/2^7$	-9.9	-12.8	-13.1	-12.8
$1/2^8$	-11.4	-12.9	-13.3	-12.6
$1/2^9$	-13.1	-12.4	-12.5	-12.5

We also carried out similar experiments for the four FEPTRKN methods feptrkn52, feptrkn73, feptrkn83, and feptrkn95. We found out from our experiments that the accuracy orders of the feptrkn52, feptrkn73, feptrkn83, and feptrkn95 methods are $p = 5$, $p = 7$, $p = 8$, and $p = 9$, respectively. Again, this agrees with our super-convergence results in Theorems 4.3 and 4.4. For simplicity we do not include the numerical results of these experiments in this paper.

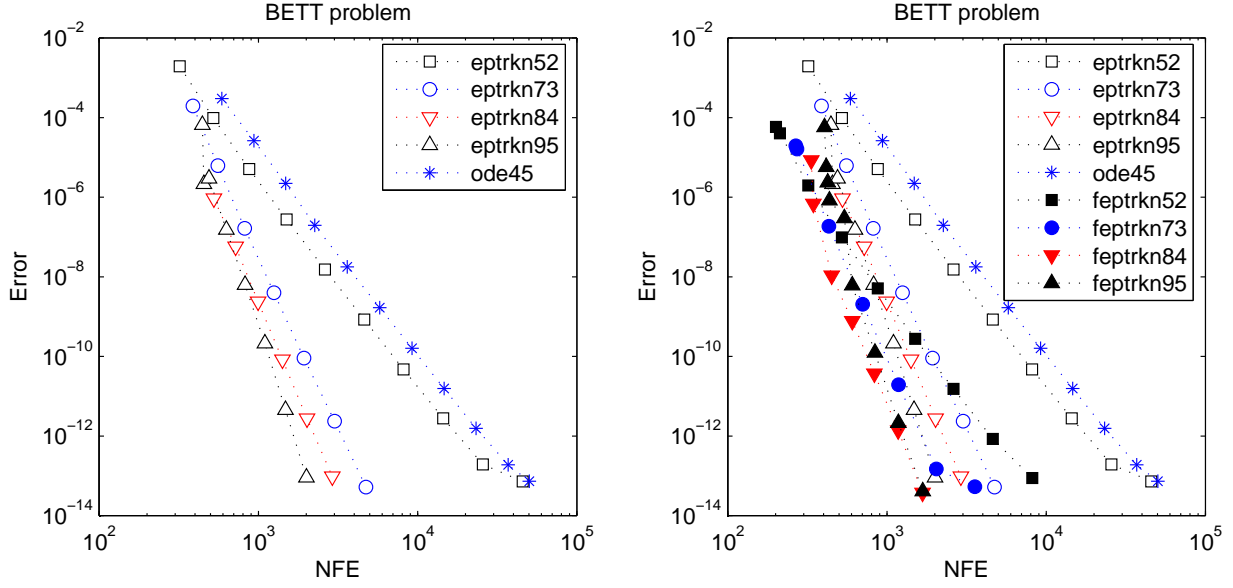


Figure 3: Numerical results for the BETT problem.

Figure 3 and Figure 4 present the numerical results for solving the BETT and NEWT problems using the eight EPTRKN and FEPTTRKN methods derived in Section 6.1. We also include numerical results obtained by the MATLAB function ode45. This function is a MATLAB implementation of the Runge-Kutta method DOPRI45 (see, e.g., [10]). The eight EPTRKN and FEPTTRKN methods are implemented with variable stepsizes technique as described in Section 5.1. In Figures 3 and 4 we denote by NFE the number times evaluating $f(t_n + c_i h_n, Y_{n,i})$ of each of these methods. Note that the main computation cost for solving large-scale equation (1) is evaluating the right-hand side function f . The errors reported in Figures 3 and 4 are computed as follows

$$\text{Error} = \sqrt{(y_1^{\text{comput}}(t_{\text{end}}) - y_1(t_{\text{end}}))^2 + (y_2^{\text{comput}}(t_{\text{end}}) - y_2(t_{\text{end}}))^2}.$$

Figure 3 plots the numerical results for the BETT problem. The left figure in Figure 3 plots the numerical results obtained by the methods eptrkn52, eptrkn73, eptrkn84, eptrkn95, and the ode45 function. One can see from this figure that all the EPTRKN methods are better than the ode45 method. The right figure in Figure 3 plots the results obtained the eight methods derived in Section 6.1 and ode45. Also, it can be seen from the right figure in Figure 3 that all of the FEPTTRKN methods are much better than their corresponding EPTRKN methods in this experiment.

Figure 4 plots the results for the NEWT problem. From the left figure in Figure 4 we can see that the eptrkn52 method is slightly better than the ode45 method. The other EPTRKN methods are much better than the ode45 method. The feptrkn95 method yields the best result for small TOL. The right figure in Figure 4 plots the results for the eight methods derived in Section 6.1 and the ode45 function. Again, we see that all functionally-fitted EPTRKN methods are superior to the corresponding EPTRKN methods. All of the methods derived in Section 6.1 except for the eptrkn52 are much better than the ode45 method when TOL is small.

Note that all computations here is done in sequential computing environment. The new methods will be much more efficient for solving large-scale nonstiff problems when implemented on shared memory computers as they require only one right-hand side function evaluation per step in parallel computing environments (see, e.g., [5]).

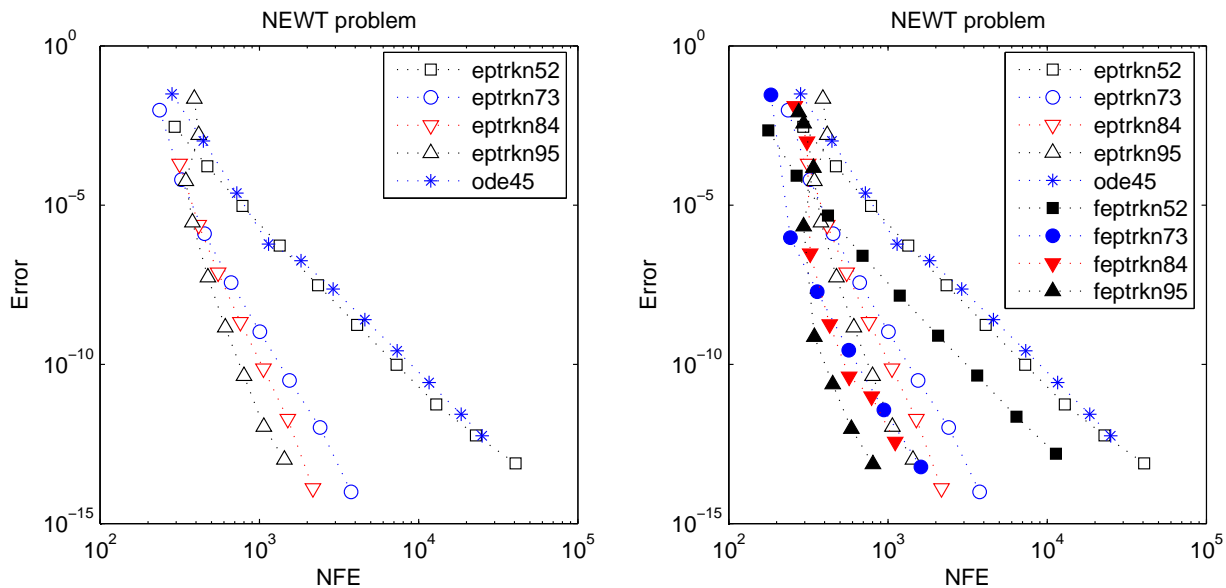


Figure 4: Numerical results for the NEWT problem when $e = 0.01$.

7 Concluding remarks

A new class of functionally fitted explicit pseudo two-step RKN (FEPTRKN) methods has been developed and studied in this paper. The advantage of functionally fitted methods over classical methods is that we can fine-tune the choice of bases to have methods which can better capture the special properties of a particular problem that may be known in advance. We have proved a new superconvergence result that allows us to construct s -stage (F)EPTRKN methods having accuracy orders $p = s + 3$ and have shown that this new super-convergence result can be realized in practice. If we choose the monomials as the basis functions, then we recover EPTRKN methods. Moreover, we can obtain a larger class of methods by choosing various basis functions such as exponential functions, trigonometric polynomials, or mixed algebraic and trigonometric polynomials. Important aspects to producing robust and efficient numerical codes, namely, variable stepsizes, error estimation via embedded methods, as well as continuous extensions are also considered. While it is primarily designed for parallel computers, numerical comparisons with the MATLAB function ode45 in the paper show that the new methods can work better than one of the most commonly used methods on nonstiff problems in which accuracy (rather than stability) controls stepsizes. The new methods generally have a competitive advantage when the basis functions and the collocation parameters are chosen suitably. Since FEPTRKN methods have the structure of EPTRKN methods, they will be more efficient when implemented on parallel computers.

References

- [1] K. Burrage, *Parallel and sequential methods for ordinary differential equations*, Oxford University Press, Oxford, 1995.
- [2] J. P. Coleman and S.C. Duxbury, Mixed collocation methods for $y'' = f(t, y)$, *J. Comput. Appl. Math.*, 126 (2000), 47–75.

- [3] N. H. Cong, K. Strehmel, R. Weiner, A general class of explicit pseudo two-step RKN methods on parallel computers, *Comput. Math. Appl.*, 38 (1999), 17-30.
- [4] N. H. Cong, Explicit pseudo two-step Runge-Kutta methods for parallel computers, *Int. J. Comput. Math.*, 73 (1999), 77-91.
- [5] N. H. Cong, H. Podhaisky, R. Weiner, Efficiency of embedded explicit pseudo two-step RKN methods on a shared memory parallel computer, *Vietnam J. Math.*, 34 (2006), no. 1, 95-108.
- [6] N. H. Cong, Explicit pseudo two-step RKN methods with stepsize control, *Appl. Numer. Math.*, 38 (2001), 135-144.
- [7] R. D'Ambrosio, M. Ferro, B. Paternoster, Trigonometrically fitted two-step hybrid methods for special second order ordinary differential equations, *Math. Comput. Simulat.*, 81 (2011), no. 5, 1068-1084,
- [8] J. M. Franco, Exponentially fitted explicit Runge-Kutta-Nyström methods, *J. Comput. Appl. Math.*, 167 (2004), 1-19.
- [9] J. M. Franco and I. Gómez, Some procedures for the construction of high-order exponentially fitted Runge-Kutta-Nyström methods of explicit type, *Comput. Phys. Commun.*, 184 (2013), 1310-1321.
- [10] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer-Verlag, Berlin, 1991.
- [11] N. S. Hoang, R. B. Sidje, and N. H. Cong, Analysis of trigonometric implicit Runge-Kutta methods, *J. Comput. Appl. Math.*, 198 (2007), 187-270.
- [12] N. S. Hoang, R. B. Sidje, N. H. Cong, On functionally-fitted Runge-Kutta methods, *BIT Numer. Math.*, 46 (2006), no. 4, 861-874.
- [13] N. S. Hoang and R. B. Sidje, On the stability of functionally-fitted Runge-Kutta methods, *BIT Numer. Math.*, 48 (2008), N1, 61-77.
- [14] N. S. Hoang and R. B. Sidje, Functionally-fitted pseudo two-step Runge-Kutta methods, *Appl. Numer. Math.*, 59 (2009), N1, 39-55.
- [15] N. S. Hoang and R. B. Sidje, Functionally-fitted Runge-Kutta-Nyström methods, (submitted for publication).
- [16] K. Ozawa, Functional fitting Runge-Kutta-Nyström method with variable coefficients, *Japan J. Indust. App. Math.*, 19 (2002), 55-85.
- [17] G. Psihoyios, T.E. Simos, Trigonometrically fitted predictor-corrector methods for IVPs with oscillating solutions, *J. Comp. Appl. Math.*, 158 (2003), no. 1, 135-144.
- [18] P. J. van der Houwen, B. P. Sommeijer, N. H. Cong, Stability of collocation-based Runge-Kutta-Nyström methods, *BIT Numer. Math.*, 31 (1991), no. 3, 469-481.